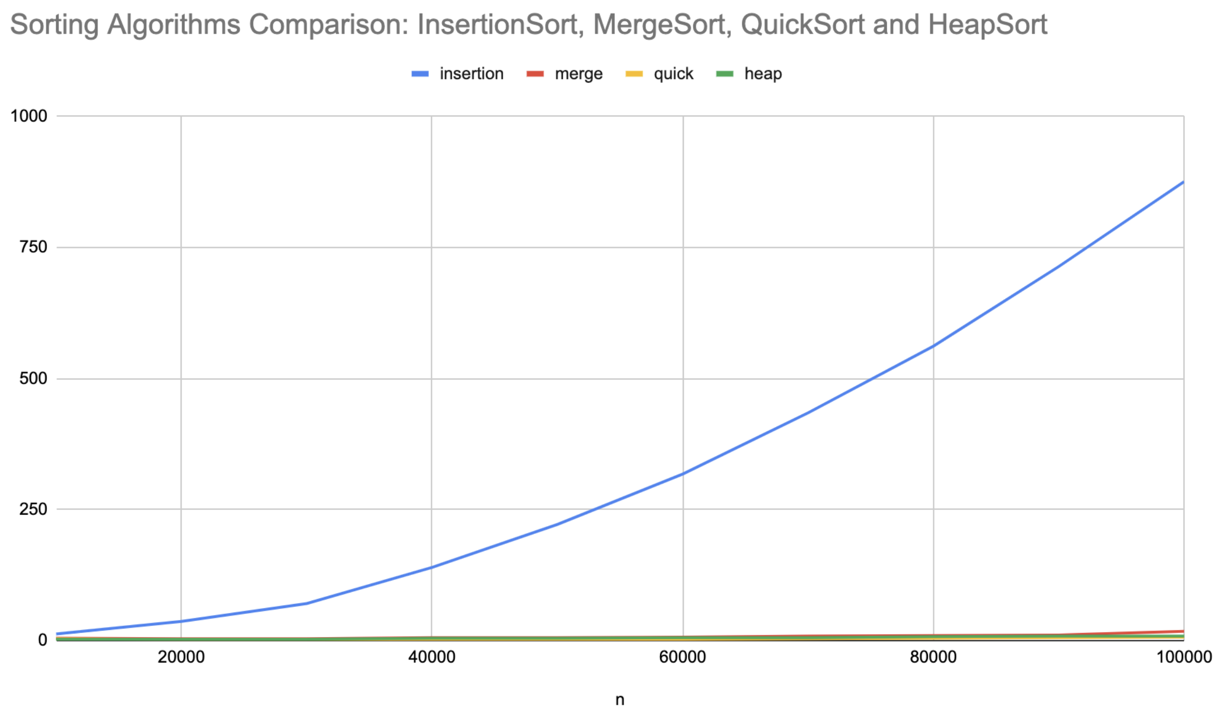
Results:

n	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
insertion	13	37	71	140	222	318	435	562	714	876
merge	5	4	4	6	6	7	9	10	11	18
quick	4	2	2	3	3	3	4	5	6	7
heap	3	3	3	5	5	6	6	8	9	9

Graph:



Observation:
As we can see from the result, insertion-sort O(n^2) takes the longest time than heap-sort, quick-sort, merge-sort and heap sort O(n log n) especially when the size of the array grows. Insertion sort is only efficient with the the number of elements is small and/or for an “almost” sorted sequence. On the other hands, merge-sort, quick-sort, and heap-sort all use a comparison-based sorting algorithm, which is at least O(n log n). Among merge-sort, quick-sort, and heap-sort, quick sort performs the best while merge-sort starts to requires longer runtime as the array size grows.

I have ran the program several time, and the result has been pretty consistent with the insertion-sort takes the longest runtime, and merge-sort, quick-sort, and heap-sort performed much better with a slightly difference among all three. Quick-sort has been performing the best so I believe the pivot has been picked well, and as the array size grows, the runtime has not grown much. I have also noticed a noticeable increase for merge-sort as the array size grows, it might be the merge part that now we have to iterate through a longer list in order to complete the sorting. Heap-sort has been in the middle as you can tell that it works well on small and/or medium-sized sequences while quick-sort has shown its quickness on runtime especially when the sizes of the array grows but not much when the size is small.

Print:

```
For Array Size 10000
Elapsed Time for Insertion Sort: 13 mils
Elapsed Time for Merge Sort: 5 mils
Elapsed Time for Quick Sort: 4 mils
Elapsed Time for Heap Sort: 3 mills

For Array Size 20000
Elapsed Time for Insertion Sort: 37 mils
Elapsed Time for Merge Sort: 4 mils
Elapsed Time for Quick Sort: 2 mils
Elapsed Time for Heap Sort: 3 mills

For Array Size 30000
Elapsed Time for Insertion Sort: 71 mils
Elapsed Time for Merge Sort: 4 mils
Elapsed Time for Quick Sort: 2 mils
Elapsed Time for Heap Sort: 3 mills

For Array Size 40000
Elapsed Time for Insertion Sort: 140 mils
Elapsed Time for Merge Sort: 6 mils
Elapsed Time for Quick Sort: 3 mils
Elapsed Time for Heap Sort: 5 mills

For Array Size 50000
Elapsed Time for Insertion Sort: 222 mils
Elapsed Time for Merge Sort: 6 mils
Elapsed Time for Quick Sort: 3 mils
Elapsed Time for Heap Sort: 5 mills

For Array Size 60000
Elapsed Time for Insertion Sort: 318 mils
Elapsed Time for Merge Sort: 7 mils
Elapsed Time for Quick Sort: 3 mils
Elapsed Time for Heap Sort: 6 mills

For Array Size 70000
```

Elapsed Time for Insertion Sort: 435 mils
Elapsed Time for Merge Sort: 9 mils
Elapsed Time for Quick Sort: 4 mils
Elapsed Time for Heap Sort: 6 mills

For Array Size 80000
Elapsed Time for Insertion Sort: 562 mils
Elapsed Time for Merge Sort: 10 mils
Elapsed Time for Quick Sort: 5 mils
Elapsed Time for Heap Sort: 8 mills

For Array Size 90000
Elapsed Time for Insertion Sort: 714 mils
Elapsed Time for Merge Sort: 11 mils
Elapsed Time for Quick Sort: 6 mils
Elapsed Time for Heap Sort: 9 mills

For Array Size 100000
Elapsed Time for Insertion Sort: 876 mils
Elapsed Time for Merge Sort: 18 mils
Elapsed Time for Quick Sort: 7 mils
Elapsed Time for Heap Sort: 9 mills

More Findings:
For Array Size 10000
Elapsed Time for Insertion Sort: 13 mils
Elapsed Time for Merge Sort: 5 mils
Elapsed Time for Quick Sort: 5 mils
Elapsed Time for Heap Sort: 5 mills

For Array Size 20000
Elapsed Time for Insertion Sort: 40 mils
Elapsed Time for Merge Sort: 7 mils
Elapsed Time for Quick Sort: 2 mils
Elapsed Time for Heap Sort: 3 mills

For Array Size 30000
Elapsed Time for Insertion Sort: 72 mils
Elapsed Time for Merge Sort: 3 mils
Elapsed Time for Quick Sort: 1 mils
Elapsed Time for Heap Sort: 4 mills

For Array Size 40000
Elapsed Time for Insertion Sort: 120 mils

```
Elapsed Time for Merge Sort: 5 mils
Elapsed Time for Quick Sort: 3 mils
Elapsed Time for Heap Sort: 5 mills

For Array Size 50000
Elapsed Time for Insertion Sort: 228 mils
Elapsed Time for Merge Sort: 7 mils
Elapsed Time for Quick Sort: 3 mils
Elapsed Time for Heap Sort: 6 mills

For Array Size 60000
Elapsed Time for Insertion Sort: 330 mils
Elapsed Time for Merge Sort: 8 mils
Elapsed Time for Quick Sort: 4 mils
Elapsed Time for Heap Sort: 6 mills

For Array Size 70000
Elapsed Time for Insertion Sort: 451 mils
Elapsed Time for Merge Sort: 9 mils
Elapsed Time for Quick Sort: 5 mils
Elapsed Time for Heap Sort: 7 mills

For Array Size 80000
Elapsed Time for Insertion Sort: 585 mils
Elapsed Time for Merge Sort: 11 mils
Elapsed Time for Quick Sort: 6 mils
Elapsed Time for Heap Sort: 8 mills

For Array Size 90000
Elapsed Time for Insertion Sort: 737 mils
Elapsed Time for Merge Sort: 13 mils
Elapsed Time for Quick Sort: 6 mils
Elapsed Time for Heap Sort: 9 mills

For Array Size 100000
Elapsed Time for Insertion Sort: 900 mils
Elapsed Time for Merge Sort: 18 mils
Elapsed Time for Quick Sort: 6 mils
Elapsed Time for Heap Sort: 10 mills
```